

Modul de lucru grafic. Biblioteci specifice modului grafic.

În mod curent, utilizatorul (programatorul) folosește modul de lucru text. Aceasta înseamnă că ecranul este văzut ca o matrice de 25 linii și 80 de coloane (este posibilă și varianta 25 linii și 40 de coloane), fiecare element având dimensiunea unui caracter.

Pentru realizarea unor desene pe ecran nu se poate folosi modul text. Este necesar să se utilizeze modul grafic, în care ecranul este văzut ca o matrice de puncte (pixeli) (840 de puncte pe linie și 640 pe coloană). Deci, cantitatea elementară de informație (image) ce poate fi afișată pe ecran este pixelul. **Fiecare element al unui desen (linie, cerc, arc) este reprezentat prin asocierea mai multor pixeli.**

Trecerea de la modul text la modul grafic necesită **inițializarea modului grafic.**

Etapele inițializării grafice sunt:

- verificarea existenței bibliotecii **graphics** și includerea acesteia în program prin directiva corespunzătoare.
- declararea variabilelor de tip pointer către tipul întreg
*int *graphdriver, int *graphmode*
- apelarea funcției *initgraph()*.

Detaliere:

- verificarea existenței bibliotecii **graphics** și includerea acesteia în program prin directiva corespunzătoare:
#include<graphics.h>

Se determină apoi poziția fișierelor (driverelor) pentru placa grafică în arborele de directoare. Acestea se afla de obicei în directorul **BGI** din structura de directoare a BorlandC (TurboC)

Ex. - în laborator această poziție este:

- **c:\borlandc\bgi;**
- **g:\tc\bgi;**

- declararea variabilelor de tip pointeri către tipul întreg
*int *graphdriver, int *graphmode* care reprezintă:

- *graphdriver* – numărul driverului grafic folosit. Acesta poate lua una din valorile:

Numele constantei	Valoare
DETECT	0
CGA	1
MCGA	2
EGA	3
EGA64	4
EGAMONO	5
IBM8514	6
HERCMONO	7
ATT400	8
VGA	9
PC3270	10

Utilizatorul nu foloseste decit arareori valorile efective a acestora.

- *graphmode* – numărul modului grafic folosit. Acesta poate lua una din valorile:

Constantă	Mod grafic	Valoare	Coloane x linii	Paleta de culori
VGA	VGA LO	0	640 x 200	16 color
	VGA MED	1	640 x 350	16 color
	VGA HI	2	640 x 480	16 color
IBM8514	IBM8514 HI	0	640 x 480	256 color
	BM8514 LO	0	1024 x 768	256 color

Determinarea valorilor pentru driverul grafic și pentru modul grafic se face, de obicei, in programe, folosind funcția **detectgraph()**

```
void detectgraph(int far *graphdriver, int far *graphmode);
```

c) - apelarea funcției *initgraph()*. Aceasta are structura;

```
initgraph(int *graphdriver, int *graphmode, char far *pathdriver);
```

Parametrii *graphdriver* și *graphmode* sunt aleși (din tabelul de la punctual a) sau determinați implicit(se apeleaza *detectgraph*).

Parametrul *pathdriver* reprezintă poziția în arborele de directoare a driverului video.(Driver video= un program care ajuta la initializarea si folosirea modului de lucru grafic)

Ex. Secvența de instrucțiuni pentru inițializarea modului grafic este:

```
int gdriver, gmode, errorcode;
```

```
char *pathdriver="c:\\borlandc\\bgi"; //sau "g:\\tc\\bgi";
```

```
detectgraph(&gdriver, &gmode);
```

```
errorcode = graphresult(); //se verifică dacă nu a apărut o eroare în
```

```
if (errorcode != grOk) // determinarea driverului sau a modului de lucru grafic
```

```
{
```

```
printf("Eroare la detectarea driverului sau modului grafic: %s\n",
```

```
grapherrormsg(errorcode));
```

```
getch();
```

```
}
```

```
initgraph(&gdriver, &gmode, pathdriver);
```

Daca în urma apelarii funcției *initgraph* apare o eroare(nu poate fi initializat modul de lucru grafic), va aparea un mesaj pe ecran. Erorile se pot datora unor cauze precum:

- imposibilitatea de a „gasi” în locația indicata fisierul **graphics.h**;

Rezolvare:

Se merge la bara de meniuri și se alege obțiunea **Options**. Apoi se alege obțiunea **Directories**, și se modifica calea corecta la secțiunea **Include directories**

- indicarea greșită a căii de acces catre driverul video.

Rezolvare:

Se merge în program la linia unde se declara variabila **char *pathdriver** si se corectează calea.

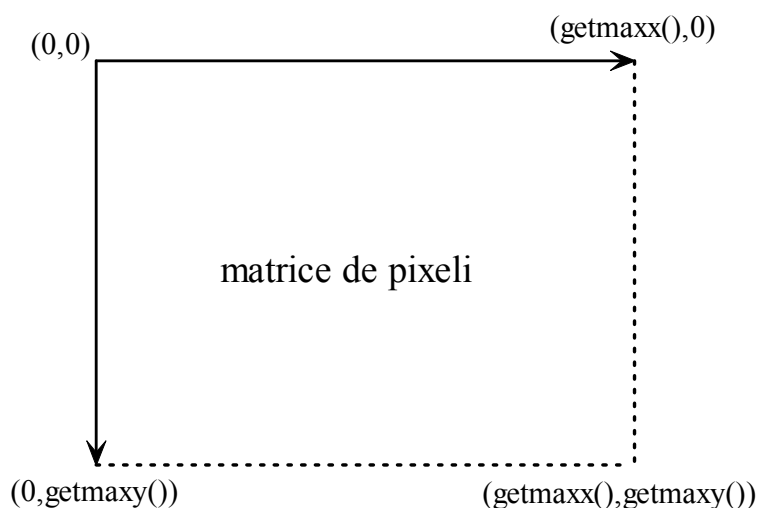
- incompatibilitatea dintre placa video si driverul ales.

Rezolvare:

*Nu are o rezolvare usoara. De obicei trebuie cautat driverul video potrivit. Pentru placile video cu mai mult de 4 MBy de memorie este posibil sa fie necesar aducerea de pe Internet a unui **pach**.*

La trecerea din mod text în mod grafic se face automat ștergerea ecranului. În mod grafic cursorul devine invizibil. Poziționarea inițială a cursorului se face în punctul din stânga sus (0,0).

Configurația ecranului în mod grafic



Colțul din stânga sus este originea axelor de coordonate (0,0). Coordonatele colțului din dreapta jos depind de driverul de placă video și de modul grafic folosit. Pentru modul grafic 640 x 480 acestea sunt (639,479), pentru modul grafic 320 x 200 acestea sunt (319,199). Aceste valori se pot însă determina și folosind funcțiile *getmaxx()* și *getmaxy()*

Funcții pentru lucrul în mod grafic

- a) - aflarea și schimbarea poziției curente pe ecran:
 - *void moverel(int dx, int dy);* //deplasează cursorul față de poziția curentă cu (dx, dy).
- b) - deplasarea cursorului în punctul P(x,y)
 - *void moveto(int x, int y);*

Desenarea elementelor grafice

- a) – funcții pentru desenarea punctelor
 - *void putpixel(int x, int y, int color);*
- b) – funcții pentru desenarea liniilor
Alegerea tipului de line:
 - *void setlinestyle(int tip_linie, unsigned odel_linie, int grosime);*

- unde *tip linie* poate avea valorile:

Constantă	Valoare	Semnificație
SOLID_LINE	0	Solid line
DOTTED_LINE	1	Dotted line
CENTER_LINE	2	Centered line
DASHED_LINE	3	Dashed line
USERBIT_LINE	4	defined line style

- grosime poate fi una din valorile:

Constantă	Valoare	Semnificație
NORM_WIDTH	1	1 pixel wide
THICK_WIDTH	3	3 pixels wide

Desenarea liniilor se realizează cu una din funcțiile:

- *void line(int x1, int y1, int x2, int y2);*
 - unde (x1,y1), (x2,y2) sunt coordonate extremităților liniei.
 - *void lineto(int x, int y);*
 - funcția desenează o linie începând cu coordonata curentă și până la coordonata de sfârșit (x,y).
 - *void linerel(int dx, int dy);*
 - funcția desenează o linie începând cu coordonata curentă, avansând pe cele două dimensiuni cu dx și dy.
- c) – funcția pentru desenarea unui dreptunghi:
- *void rectangle(int left, int top, int right, int bottom);*
 - funcția desenează un dreptunghi cu laturile paralele cu axele de coordonate. Punctele (x1,y1) respectiv (x2,y2) reprezintă coordonatele din colțurile stânga sus, respectiv dreapta jos ale dreptunghiului.
- d) – funcția pentru desenarea unui poligon închis:
- *void drawpoly(int numpoints, int *polypoints);*
 - funcția primește ca argumente nr de colțuri ale poligonului și un tablou de coordonate ale acestor puncte.
- e) – funcții pentru desenarea curbilor:
- *void circle(int x, int y, int radius);*
 - funcția desenează un cerc cu centrul în punctul de coordonat (x,y) și de rază radius;
 - *void arc(int x, int y, int stangle, int endangle, int radius);*
 - funcția desenează un arc de cerc cu centrul în punctul de coordonat (x,y), deschidere între cele două unghiuri și de rază radius;
 - *void ellipse(int x, int y, int stangle, int endangle, int xradius, int yradius);*
 - funcția desenează o porțiune dintr-o elipsă cu centrul în punctul de coordonat (x,y), deschidere între cele două unghiuri și de razele xradius și yradius;

- f) – funcții pentru scrierea unui text în mod grafic:
- *void outtextxy(int x, int y, char *textstring);*
 - funcția scrie pe ecran începând cu punctul de coordonate (x,y) textul care este memorat în stringul *textstring*.
- g) – funcții cu utilizare generală:
- *void cleardevice(void);*
 - funcție pentru ștergerea ecranului în mod grafic.
 - *getmaxx(), getmaxy()*
 - funcțiile permit determinarea numărului maxim de pixeli pe abscisă respectiv ordonată.
 - *closegraph();*
 - închiderea sesiunii de lucru în modul grafic și revenirea automată în modul de lucru text.

Observatie !

closegraph() este obligatorie la terminarea desenarii. Problema care apare este ca se inchide atit de repede ca nu mai este timp pentru a vedea ce s-a desenat. De aceea, se recomanda folosirea instructiunii *getch()* inaintea instructiunii *closegraph()*.

Mai jos sunt toate functiile modului grafic. Pentru a vedea ce fac si cum se utilizeaza scrieti numele in functiei in C , va positionati sub prima litera a acestuia si apoi apasati combinatia de taste **CTRL+F1**

arc	getgraphmode	imagesize	setactivepage
bar	getimage	initgraph	setallpalette
bar3d	getlinesettings	installuserdriver	setaspectratio
circle	getmaxcolor	installuserfont	setbkcolor
cleardevice	getmaxmode	line	setcolor
clearviewport	getmaxx	linerel	setfillpattern
closegraph	getmaxy	lineto	setfillstyle
detectgraph	getmodename	moverel	setgraphbufsize
drawpoly	getmoderange	moveto	setgraphmode
ellipse	getpalette	outtext	setlinestyle
fillellipse	getpalettesize	outtextxy	setpalette
fillpoly	getpixel	pieslice	setrgbpalette
floodfill	gettextsettings	putimage	settextjustify
getarcoords	getviewsettings	putpixel	settextstyle
getaspectratio	getx	rectangle	setusercharsize
getbkcolor	gety	registerbgidriver	setviewport
getcolor	graphdefaults	registerfarbgidriver	setvisualpage
getdefaultpalette	grapherrormsg	registerbgifont	setwritemode
getdrivername	_graphfreemem	registerfarbgifont	textheight
getfillpattern	_graphgetmem	restorecrtmode	textwidth
getfillsettings	graphresult	sector	

Laborator C_4 - Programarea Calculatoarelor

Exercițiu:

Să se reprezinte funcția $f(x)$ dată sub formă de tabel:

$x[9]=\{1,2,3,4,5,6,7,8,9\};$

$f[9]=\{10,20,30,40,50,60,70,80,90\};$

Etapele generale implementării algoritmului:

1. - determinarea tabloului funcției x , $f(x)$. (Această etapă ne este parcursă dacă funcția este deja dată sub forma $x[]$, $f[]$);
2. - determinarea elementelor maxim și minim din valorile funcției ($f[]$)
3. - inițializarea modului grafic
4. - calcularea coeficientului de scară pentru cazul cind desenul ar depasi ecranul
 $kx=(getmaxx()-30)/xmax-xmin;$
 $ky=(getmaxy()-30)/fmax-fmin;$
5. - calcularea valorii xecran, yecran (valori translate cu coeficientul de scară)
 $xecran[i]=(xi-xmin)*kx;$
 $yecran[i]=(f[i]-fmin)*ky;$
6. - calculul valorii fecran[i]
 $fecran[i]=getmaxy()-yecran[i]$
7. - se alege modul de reprezentare (puncte, segmente de dreaptă, histogramă)
8. - trasare și afișare.

Listingul programului este:

```
#include <graphics.h>
#include <stdio.h>
#include <process.h> //pentru detectarea erorilor
#include <conio.h>

int x[]={1,2,3,4,5,6,7,8,9};
int f[]={10,20,30,40,50,60,70,80,90};
int xmax,xmin,fmin,fmax,xecran[10],fecran[10];
float kx,ky;

void main()
{
    int gdriver, gmode, errorcode;
    char* cale="c:\\borlandc\\bgi";
    /* detectia driverului grafic si a modului */
    detectgraph(&gdriver, &gmode);
    /* test daca a aparut o eroare la detectia modului
    grafic*/
    errorcode = graphresult();
    if (errorcode != grOk) /* s-a gasit o eroare */
    {
        printf("Graphics error: %s\n",
grapherrormsg(errorcode));
        printf("Eroare la initializarea modului grafic:");
        getch();
        exit(1); /*iesire din program */
    }
    /* initializarea modului grafic */
    initgraph(&gdriver, &gmode, cale);

    /*afierea maximului si minimului*/
    xmax=xmin=x[0];
    for(int i=0;i<9;i++)
    {
        if(xmax<x[i])
            xmax=x[i];
        if(xmin>x[i])
            xmin=x[i];
    }
    fmax=fmin=f[0];
    for(int i=0;i<9;i++)
    {
        if(fmax<f[i])
            fmax=f[i];
        if(fmin>f[i])
            fmin=f[i];
    }
    /*aducerea la scara pentru ca desenul sa nu depaseasca
    ecranul*/
    kx=(getmaxx())/(xmax-xmin);
    ky=(getmaxy()-20)/(fmax-fmin);
    /*desenarea efectiva a fiecarui punct*/
    for(int i=0;i<9;i++)
    {
        xecran[i]=(x[i]-xmin)*kx;
        fecran[i]=getmaxy()-((f[i]-fmin)*ky);
        printf("%d ",fecran[i]);
        line(xecran[i],getmaxy(),xecran[i],fecran[i]);

        //sau bar(xecran[i],fecran[i],xecran[i]+kx,fecran[i]+ky);
    }
    getch();
    /* parasirea modului grafic*/
    closegraph();
}
```

Laborator C_4 - Programarea Calculatoarelor

Exercițiu:

Să se reprezinte funcția $f(x)=x^2-5x+3$ data prin n valori citite de la tastatura

Etapele implementării algoritmului:

1. - determinarea tabloului funcției x, f(x). Daca este cazul, vectorul x[] se ordoneaza inainte de calculul functiei f() $=y[]$ (vaorile functiei in punctele x sunt trecute in vectorul y).
2. - determinarea elementelor minim din vectori pentru a face functia sa “încapa” in ecran(mutarea axelor de coordonate)
 $x[]=x[]-xmin$
 $y[]=y[]-ymin$
3. – inițializarea modului grafic
4. - calcularea coeficientului de scară pentru cazul cind desenul ar depasi ecranul
 $kx=(getmaxx())/xmax;$
 $ky=(getmaxy())/ymax;$
5. – calcularea valorii xecran, yecran (valori translate cu coeficientul de scară)
 $x[]=x[]*kx;$
 $y[]=y[]*ky;$
6. – calculul valorii yecran[i]
 $yecran[i]=getmaxy()-y[]$
7. – trasare și afișare.

Listingul programului este:

```
#include <graphics.h>
#include <stdio.h>
#include <conio.h>
void main()
{
    int gd, gm;
    float x[20];
    float y[20];
    int n,xmax,xmin,ymin,ymax,i,j;
    float kx,ky;
    float aux;
    printf("dati o var. oarecare");
    scanf("%f",&aux);
    clrscr();

    /* detectia driverului grafic si a modului */
    gd=DETECT;
    detectgraph(&gdriver, &gmode);

    //citirea vectorului x
    printf("dati n");
    scanf("%d",&n);
    for(int i=0;i<n;i++)
        scanf("%f",x[i]);

    /* ordonarea vectorului x */
    for(int i=0;i<n-1;i++)
        for(int j=i+1;j<n;j++)
            if(x[i]>x[j])
            {
                aux=x[i];
                x[i]=x[j];
                x[j]=aux;
            }

    //calcularea vectorului y
    for(int i=0;i<n-1;i++)
        y[i]=x[i]*x[i]-5*x[i]+3;

    /* initializarea modului grafic */
    initgraph(&gdriver, &gmode, cale);

    /* aflarea maximului si minimului */
    xmax=xmin=x[0];
    for(int i=0;i<n;i++)
    {
        if(xmax<x[i])
            xmax=x[i];
        if(xmin>x[i])
            xmin=x[i];
    }
    ymax=ymin=y[0];
    for(int i=0;i<n;i++)
    {
        if(ymax<y[i])
            ymax=y[i];
        if(ymin>y[i])
            ymin=y[i];
    }

    // mutarea originii
    for(int i=0;i<n;i++)
    {
        x[i]=x[i]-xmin;
        y[i]=y[i]-ymin;
    }
}
```

Laborator C_4 - Programarea Calculatoarelor

```
/*aducerea la scara pentru ca desenul sa nu //rasturnarea imaginii
depaseasca ecranul*/
for(int i=0;i<n;i++)
    y[i]=getmaxy()-y[i];
kx=(getmaxx())/xmax;
ky=(getmaxy())/ymax;
for(int i=0;i<n;i++)
{
    x[i]=x[i]*kx;
    y[i]=y[i]*ky;
}

/*desenarea efectiva a fiecarui punct*/
for(int i=1;i<n;i++)
line(x[i-1],y[i-1],x[i],y[i]);
getch();
/* parasirea modului grafic*/
closegraph();
}
```

Exercițiu:

Se considera un vector de 3 elemente intregi reprezentand procentajul candidaților din cursa prezidențială la un sondaj de opinie. Să se reprezinte o histogramă, pe baza datelor acestora.

Fie $y[3]$ vectorul respectiv.

Etapele generale sunt aceleași:

1. – citirea vectorului y . Întrucât axa Ox este data de dimensiunea ecranului, nu mai avem ce calcula pentru axa Ox. NU trebuie ordonat acest vector.
2. — ~~determinarea elementelor maxim și minim din valorile funcției ($x[i]$)~~ Întrucât procentele nu pot fi decât numere pozitive, nu vom avea motive să mutăm originea! Aceasta înseamnă că această etapă se suprimă.
3. – inițializarea modului grafic
4. - calcularea coeficientului de scară pentru cazul când desenul ar depăși ecranul
 kx = nu mai este cazul, dar calculăm un $h = \text{getmaxx}()/3$, reprezentând dimensiunea fiecăre din cele 3 histograme pe axa Ox.
 $ky = (\text{getmaxy}())/ymax$; (bineînțeles că se calculează mai întâi maximul din y).
5. – calcularea valorii x_{ecran} , y_{ecran} (valori translate cu coeficientul de scară)
 $y[i] = (y[i]) * ky$;
6. – rasturnarea imaginii
 $y[i] = \text{getmaxy}() - y[i]$
7. – trasare și afișare.

Listiugul programului este:

```
#include <graphics.h>
#include <stdio.h>
#include <conio.h>
void main()
{
    int gd, gm;
    float y[20];
    int ymin, ymax, i, j;
    float h, ky;
    float aux;
    printf("dati o var. oarecare");
    scanf("%f", &aux);
    clrscr();
    /* detectia driverului grafic si a modului */
    gd = DETECT;
    detectgraph(&gdriver, &gmode);
    //citirea vectorului y
    for(int i=0; i<3; i++)
        scanf("%f", &y[i]);
    /* initializarea modului grafic */
    initgraph(&gdriver, &gmode, &ale);

    /* aflarea maximului */
    ymax = y[0];
    for(int i=0; i<3; i++)
        if(ymax < y[i])
            ymax = y[i];
    /*aducerea la scara pentru ca desenul sa nu depaseasca ecranul*/
    h = (getmaxx())/3;
    ky = (getmaxy())/ymax;
    for(int i=0; i<3; i++)
        y[i] = y[i] * ky;
    //rasturnarea imaginii
    for(int i=0; i<3; i++)
        y[i] = getmaxy() - y[i];
    /*desenarea efectiva a fiecarui punct*/
    for(int i=0; i<3; i++)
        bar(i * h, getmaxy() - (i + 1) * h, y[i] * 50);
    getch();
    /* parasirea modului grafic */
    closegraph();
}
```